## The Sudbury Neutrino Observatory: Monitoring the Experiment

Y.D. Chan, M.C. Isaac, M. Dragowsky, K.T. Lesko, E.B. Norman, A. Schuelke<sup>†</sup> and R.G. Stokstad

Unlike most accelerator-based experiments, the Sudbury Neutrino Observatory (SNO) will be operated 24-hours a day for a planned duration of 5 - 10 years. It is, therefore, essential to develop a flexible and reliable data monitoring program to provide up-to-date information about the status of the experiment, for SNO participants both on-site or off-site.

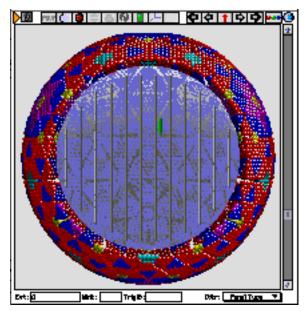


Fig. 1. SNO monitoring object CM\_PSUP.

The scope of SNO monitoring covers two areas: (1) the past and present status of the detector/cavity hardware, and (2) the status of data acquisition (DAQ). There are two classes of tasks for DAQ monitoring. The Low-Level tasks are performed by PowerPC CPUs located underground inside the SNO cavity. These desktop computers control the DAQ task and initialize the operation of the ~20 embedded CPU's in the SNOCrates. The main tasks are component monitoring, diagnostics, conditional alert, and supernovae watch. An object-oriented SNO DAQ C++ Framework has been developed over the years by the SNO DAQ group. In particular, a SNOObject class forms the basis of all monitoring objects. All modules can be preconfigured or set up dynamically during run time. Fig.1 shows an example of a detector geometry access object connected to a data source.

High-Level monitoring tasks performed on UNIX workstations situated in the SNO surface building. The main tasks performed here include acquisition statistics, long term stability evaluation, as well as first-pass event fitting and histogram analysis. In compliance with coding standards within the collaboration, a stand-alone monitoring program, SNOMON, has been developed to include features from both the official SNO Monte Carlo code, SNOMAN, and tools from the versatile CERNLIB application, PAW++. Streams of DAQ data filtered native are CERNLIB/ZEBRA format before they being processed by SNOMON. Fig. 2 shows the familiar PAW++-like user environment for SNOMON.

The surface workstations also serve as hosts for off-site monitoring. Techniques and tools from the WWW community, including Java-based dynamic content update and event browsers, are employed.

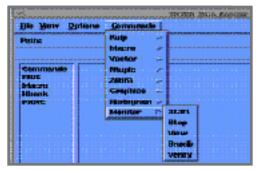


Fig. 2. SNO monitoring program SNOMON.

## Footnotes and References

†‡Die Arbeit wurde mit Unterstützung eines Stipendiums im Rahmen des Gemeinsamen Hochschulsonder-programms III von Bund and Ländern über den DAAD ermöglicht.